Adam Gaudreau, Ashley Hale, Tim Larocque

Mobile App Programming I

Guanling Chen

8 December 2016

Team Submission

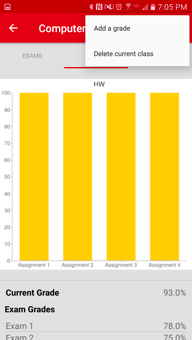
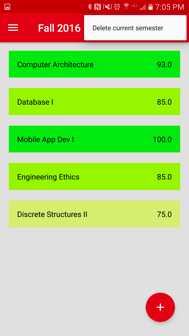
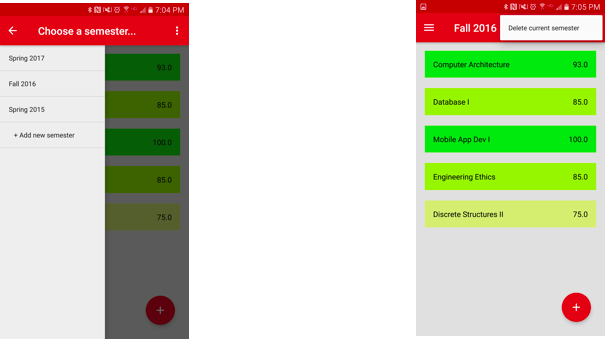
**Project Goal**

Getting straight A’s through any academic institution is hard enough, so being unsure of your current standing in each class can only make things more difficult. Since most classes, especially in college, don’t have a system to keep track of your grades, you may go through the whole semester without knowing how you are doing in that class. In addition, keeping track of all returned homeworks, quizzes, and exams can be a nightmare. Our android application, Keep the Grade, seeks to alleviate some of the stress caused by uncertainty on academic standings by providing a simple, user-friendly interface to keep track of all your previous and current grades.

**Project Features**

* Database:
  + Database consists of four tables:
    - Semesters -- Contains semester season and year
    - Classes -- Contains class grade and name
    - Grades -- Contains type, name, and grade
    - Weights -- Contains the weight of each type of Grade
  + Semesters and Classes are both used to display data in MainActivity.
  + Grades is used to display data in ClassActivity.
  + Weights is used alongside Grades when calculating a class’s final grade.
* Semester:
  + Can add (or delete) a semester, even if the semester has already passed
    - If the semester has passed, it will ask for the final grade of every class in that semester instead of the full class details
    - If the semester has not passed, for each class created in that semester it will ask for the class’s name and the weights on each of the sections of that class (i.e. Exam, HW, Quiz, Final Exam)
  + Color codes the class depending on the current grade of that class
    - Red shade = failing ( < C-)
    - Yellow shade = in danger of failing (C-, C, C+)
    - Green shade = grade is not failing nor in danger of failing (B - A)
* Class:
  + User can set the weights of each of the sections (Exam, HW, Quiz, Final Exam) on class creation
    - Weights must add up to 100% (can’t do extra credit)
  + Able to add grades to the Exam, HW, Quiz, or Final Exam sections
  + Can delete grades by tapping on the specific grade the user wants removed
  + Updates the expected final grade of the class each time the grades are updated
  + Shows a chart to give a visual representation of the grades for three of the sections (i.e. Exam, HW, and Quiz)
  + Can delete the entire class in the options menu of that class
* Chart:
  + Three sections:
    - Exams
    - HW (Homework)
    - Quizzes
  + No chart for Final Exam because there is only one grade
  + Each chart shows a bar graph representation of each of the grades for each section (i.e. Exams show Exam 1, Exam 2, etc.)
  + The bars are scalable
    - The more grades there are in the section, the thinner the bar is for each grade
  + On the X-axis, it shows the name of the Exam/HW/Quiz instead of a numeric value (i.e. names like Bio Exam, Midterm, Exam 3, instead of 1, 2, 3, etc.)
  + If a grade is added or deleted the chart will update when the user exits out of the class and enters again

**Project Design**



On the left is the drawer containing the semesters that the user has inputted. The drawer also allows the user to add a new semester to the drawer. The new semester can be the new current semester or an old semester with completed classes. With the drawer closed (the center image), the user can see all of his/her classes at a glance. The classes are displayed with the color code described above. When the user selects a class, he/she is brought to the detailed display of the class. In here the user can add and delete grades or even delete the class as a whole.

**File Structure**

The following section describes what each file does, as well as who contributed to it.

Adam

|  |  |
| --- | --- |
| File | Description |
| Class.java | The class object holds all relevant information for a class. |
| ClassAdapter.java | Displays each class element in a nice ListView. |
| DatabaseContract.java | A set of defined Strings and Integers to make database queries consistent. |
| DatabaseHelper.java | Ensures that a database is created when user first runs the app. |
| DatabaseUtils.java | Where all the database insertions, deletions, and queries are made. |
| MainActivity.java | The activity that opens when app is launched. Displays current classes in a semester, as well as a drawer and options menu. |
| Semester.java | The semester object holds all relevant information for a semester. |
| AddGradeActivity.java | A form to add a grade to a certain class. |
| Grade.java | The grade object holds all relevant information for a grade. |
| GradeAdapter.java | Displays each grade element in a ListView. |
| Weight.java | The weight object holds all relevant info on class weights. |

Tim

|  |  |
| --- | --- |
| File | Description |
| ClassActivity.java | The activity that opens when the user selects a class from the Main Activity. Displays grades for the class in both a chart and a list, as well as an options menu. |
| MainActivity.java | The activity that opens when app is launched. Displays current classes in a semester, as well as a drawer and options menu. |
| activity\_class.xml | The layout for the Class Activity. Each widget is contained in an overall linear view. Grades are displayed in a listview and the charts are in a tabular layout. |
| activity\_add\_grade.xml | The layout for the Add Grade Activity. A form. |
| activity\_main.xml | The layout for the Main Activity. |
| add\_class\_popup.xml | The layout for the popup that appears for adding a class in Main Activity when the semester is current. |
| add\_class\_popup\_completed.xml | The layout for the popup that appears for adding a class in Main Activity when the semester has passed. |
| class\_item.xml | The layout for items contained in the listview of the Main Activity. |
| drawer\_item.xml | The layout for items contained in the drawer in the Main Activity. |
| grade\_item.xml | The layout for items contained in a listview in the Class Activity. |

Ashley

|  |  |
| --- | --- |
| File | Description |
| CategoryAdapter.java | Displays each Chart from ChartAdapter.java in a fragment |
| Chart.java | Creates a Chart object using info passed in by CategoryAdapter.java |
| ChartAdapter.java | Displays a Chart |
| ChartFragment.java | Makes a fragment for each Chart |
| ClassActivity.java | Creates a ViewPager to be able to swipe between Chart fragments and tabs for each fragment |
| MySingleton.java | Design pattern that restricts the instantiation of a Chart class to one object |
| item\_list.xml | Shows a NetworkImageView of the Chart object |